

## Ähnlich klingende Wörter

### Themen

Mit dieser Aufgabe wollen wir Folgendes trainieren:

- Strings und Stringverarbeitung

### Beschreibung

Mit der phonetischen Suche können wir nach ähnlich klingenden Wörtern und Lauten in den menschlichen Sprachen suchen. Dabei werden wir versuchen, eine phonetische („klangliche“) Repräsentation eines Wortes zu finden und nach dieser zu suchen. Zwei Wörter „passen“ dann zueinander, wenn sie die gleiche phonetische Repräsentation aufweisen.

Der Match Rating Approach (MRA) ist ein phonetisches Suchverfahren, welches wir in dieser Aufgabe programmieren wollen.

Zur Umsetzung werden wir zwei Funktionen schreiben:

- `isVowel()` soll prüfen, ob ein Zeichen ein Vokal ist. Vokale sind die Zeichen 'A', 'E', 'I', 'O' und 'U'.
- `mra()` setzt den MRA um. Der MRA-Algorithmus
  - wandelt alle Buchstaben in Grossbuchstaben um,
    - reduziert Doppelkonsonanten auf ein Zeichen (Konsonanten sind alle Zeichen, die keine Vokale sind) und
  - entfernt alle Vokale aus dem Wort, ausser wenn ein Vokal am Anfang des Wortes steht.

Wenn die Länge des phonetischen MRA-Wortes länger als 6 Zeichen ist, wird es auf die ersten und die letzten drei Zeichen reduziert.

### Aufgabenstellung

Programmiere die zwei beschriebenen Funktionen und setze damit den Match Rating Approach (MRA) um.

Testfälle

- "Basketball" → "BSKTBL"
- "Armbanduhr" → "ARMDHR"
- "Programm" → "PRGRM"
- "Algorithmuschreiber" → "ALGRBR"

### Algorithmische Tipps

Wenn du stockst und nicht weiterweisst, dann versuch mal Folgendes:

- Zur Umwandlung von Strings in Grossbuchstaben existieren integrierte Funktionen in Java (`toUpperCase()`).

- Du kannst alle Zeichen in einer Schleife durchgehen und diese auf Vokale und die Position prüfen. Durch die Iterationsvariable weißt du, an welcher Stelle des Strings du dich gerade befindest. Ebenso kannst du das aktuelle Zeichen mit dem zuletzt hinzugefügten Zeichen abgleichen. Nach erfolgter Prüfung kannst du das jeweilige Zeichen dem Ergebnisstring hinzufügen.
- Ein sehr hilfreicher Befehl in dieser Aufgabe ist `continue`. Damit kannst du in Schleifen die Ausführung des aktuellen Iterationsschritts abbrechen und in die nächste Iteration springen. Unter Umständen kann der Einsatz dieses Befehls aufwendig verschachtelte `if`-Anweisungen ersparen.